Filtering Duplicates

duplicates

As files are shared on the internet, they are often resized, cropped, converted to a different format, altered by the original or a new artist, or turned into a template and reinterpreted over and over and over. Even if you have a very restrictive importing workflow, your client is almost certainly going to get some **duplicates**. Some will be interesting alternate versions that you want to keep, and others will be thumbnails and other low-quality garbage you accidentally imported and would rather delete. Along the way, it would be nice to merge your ratings and tags to the better files so you don't lose any work.

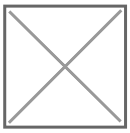
Finding and processing duplicates within a large collection is impossible to do by hand, so I have written a system to do the heavy lifting for you. It currently works on still images, but an extension for gifs and video is planned.

Hydrus finds *potential* duplicates using a search algorithm that compares images by their shape. Once these pairs of potentials are found, they are presented to you through a filter like the archive/delete filter to determine their exact relationship and if you want to make a further action, such as deleting the 'worse' file of a pair. All of your decisions build up in the database to form logically consistent groups of duplicates and 'alternate' relationships that can be used to infer future information. For instance, if you say that file A is a duplicate of B and B is a duplicate of C, A and C are automatically recognised as duplicates as well.

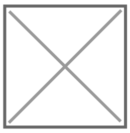
This all starts on--

the duplicates processing page

On the normal 'new page' selection window, hit *special->duplicates processing*. This will open this page:



Let's go to the preparation page first:



The 'similar shape' algorithm works on *distance*. Two files with 0 distance are likely exact matches, such as resizes of the same file or lower/higher quality jpegs, whereas those with distance 4 tend to be to be hairstyle or costume changes. You will be starting on distance 0 and not expect to ever go above 4 or 8 or so. Going too high increases the danger of being overwhelmed by false positives.

If you are interested, the current version of this system uses a 64-bit phash to represent the image shape and a VPTree to search different files' phashes' relative hamming distance. I expect to extend it in future with multiple phash generation (flips, rotations, and 'interesting' image crops and video frames) and most-common colour comparisons.

Searching for duplicates is fairly fast per file, but with a large client with hundreds of thousands of files, the total CPU time adds up. You can do a little manual searching if you like, but once you are all settled here, I recommend you hit the cog icon on the preparation page and let hydrus do this page's catch-up search work in your regular maintenance time. It'll swiftly catch up and keep you up to date without you even thinking about it.

Start searching on the 'exact match' search distance of 0. It is generally easier and more valuable to get exact duplicates out of the way first.

Once you have some files searched, you should see a potential pair count appear in the 'filtering' page.

the filtering page

Processing duplicates can be real trudge-work if you do not set up a workflow you enjoy. It is a little slower than the archive/delete filter, and sometimes takes a bit more cognitive work. For many users, it is a good task to do while listening to a podcast or having a video going on another screen.

If you have a client with tens of thousands of files, you will likely have thousands of potential pairs. This can be intimidating, but do not worry--due to the A, B, C logical inferrences as above, you will not have to go through every single one. The more information you put into the system, the faster the number will drop.

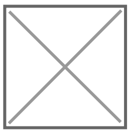
The filter has a regular file search interface attached. As you can see, it defaults to *system:everything*, but you can limit what files you will be working on simply by adding new search predicates. You might like to only work on files in your archive (i.e. that you know you care about to begin with), for instance. You can choose whether both files of the pair should match the search, or just one. 'creator:' tags work very well at cutting the search domain to something more manageable and consistent--try your favourite creator!

If you would like an example from the current search domain, hit the 'show some random potential pairs' button, and it will show two or more files that seem related. It is often interesting and surprising to see what it finds! The action buttons below allow for quick processing of these pairs and groups when convenient (particularly for large cg sets with 100+ alternates), but I recommend you leave these alone until you know the system better.

When you are ready, launch the filter.

the duplicates filter

We have not set up your duplicate 'merge' options yet, so do not get too into this. For this first time, just poke around, make some pretend choices, and then cancel out and choose to forget them.



Like the archive/delete filter, this uses quick mouse-clicks, keyboard shortcuts, or button clicks to action pairs. It presents two files at a time, labelled A and B, which you can quickly switch between just as in the normal media viewer. As soon as you action them, the next pair is shown. The two files will have their current zoom-size locked so they stay the same size (and in the same position) as you switch between them. Scroll your mouse wheel a couple of times and see if any obvious differences stand out.

Please note the hydrus media viewer does not currently work well with large resolutions at high zoom (it gets laggy and may have memory issues). Don't zoom in to 1600% and try to look at jpeg artifact differences on very large files, as this is simply not well supported yet.

The hover window on the right also presents a number of 'comparison statements' to help you make your decision. Green statements mean this current file is probably 'better', and red the opposite. Larger, older, higher-quality, more-tagged files are generally considered better. These statements have scores associated with them (which you can edit in *file->options->duplicates*), and the file of the pair with the highest score is presented first. If the files are duplicates, you can *generally* assume the first file you see, the 'A', is the better, particularly if there are several green statements.

The filter will need to occasionally checkpoint, saving the decisions so far to the database, before it can fetch the next batch. This allows it to apply inferred information from your current batch and reduce your pending count faster before serving up the next set. It will present you with a quick interstitial 'confirm/back' dialog just to let you know. This happens more often as the potential count decreases.

the decisions to make

There are three ways a file can be related to another in the current duplicates system: duplicates, alternates, or false positive (not related).

False positive (not related) is the easiest. You will not see completely unrelated pairs presented very often in the filter, particularly at low search distances, but if the shape of face and hair and clothing happen to line up (or geometric shapes, often), the search system may make a false positive match. In this case, just click 'they are not related'.

Alternate relations are files that are not duplicates but obviously related in some way. Perhaps a costume change or a recolour. Hydrus does not have rich alternate support yet (but it is planned, and highly requested), so this relationship is mostly a 'holding area' for files that we will revisit for further processing in the future.

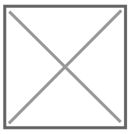
Duplicate files are of **the exact same thing**. They may be different resolutions, file formats, encoding quality, or one might even have watermark, but they are fundamentally different views on the exact same art. As you can see with the buttons, you can select one file as the 'better' or say they are about the same. If the files are basically the same, there is no point stressing about which is 0.2% better--just click 'they are the same'. For better/worse pairs, you might have reason to keep both, but most of the time I recommend you delete the worse.

You can customise the shortcuts under *file->shortcuts->duplicate_filter*. The defaults are:

- Left-click or space: **this is better, delete the other**.
- Right-click: **they are related alternates**.
- Middle-click: **Go back one decision.**
- Enter/Escape: **Stop filtering.**

merging metadata

If two duplicates have different metadata like tags or archive status, you probably want to merge them. Cancel out of the filter and click the 'edit default duplicate metadata merge options' button:



By default, these options are fairly empty. You will have to set up what you want based on your services and preferences. Setting a simple 'copy all tags' is generally a good idea, and like/dislike ratings also often make sense. The settings for better and same quality should probably be similar, but it depends on your situation.

If you choose the 'custom action' in the duplicate filter, you will be presented with a fresh 'edit duplicate merge options' panel for the action you select and can customise the merge specifically for that choice. ('favourite' options will come here in the future!)

Once you are all set up here, you can dive into the duplicate filter. Please let me know how you get on with it!

what now?

The duplicate system is still incomplete. Now the db side is solid, the UI needs to catch up. Future versions will show duplicate information on thumbnails and the media viewer and allow quick-navigation to a file's duplicates and alternates.

For now, if you wish to see a file's duplicates, right-click it and select *file relationships*. You can review all its current duplicates, open them in a new page, appoint the new 'best file' of a duplicate group, and even mass-action selections of thumbnails.

You can also search for files based on the number of file relations they have (including when setting the search domain of the duplicate filter!) using *system:file relationships*. You can also search for best/not best files of groups, which makes it easy, for instance, to find all the spare duplicate files if you decide you no longer want to keep them.

I expect future versions of the system to also auto-resolve easy duplicate pairs, such as clearing out pixel-for-pixel png versions of jpgs.

game cgs

If you import a lot of game CGs, which frequently have dozens or hundreds of alternates, I recommend you set them as alternates by selecting them all and setting the status through the thumbnail right-click menu. The duplicate filter, being limited to pairs, needs to compare all new members of an alternate group to all other members once to verify they are not duplicates. This is not a big deal for alternates with three or four members, but game CGs provide an overwhelming edge case. Setting a group of thumbnails as alternate 'fixes' their alternate status immediately, discounting the possibility of any internate duplicates, and provides an easy way out of this situation.

more information and examples

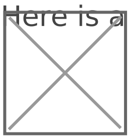
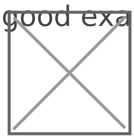
better/worse

Which of two files is better? Here are some common reasons:

- higher resolution
- better image quality
- png over jpg for screenshots
- jpg over png for busy images
- jpg over png for pixel-for-pixel duplicates
- a better crop
- no watermark or site-frame or undesired blemish
- has been tagged by other people, so is likely to be the more 'popular'

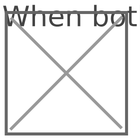
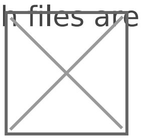
However these are not hard rules--sometimes a file has a larger resolution or filesize due to a bad upscaling or encoding decision by the person who 'reinterpreted' it. You really have to look at it and decide for yourself.

Here is a good example of a better/worse pair:

The first image is better because it is a png (pixel-perfect pngs are always better than jpgs for screenshots of applications--note how obvious the jpg's encoding artifacts are on the flat colour background) and it has a slightly higher (original) resolution, making it less blurry. I presume the second went through some FunnyJunk-tier trash meme site to get automatically cropped to 960px height and converted to the significantly smaller jpeg. Whatever happened, let's drop the second and keep the first.

When both files are jpgs, differences in quality are very common and often significant:

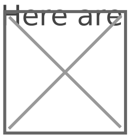
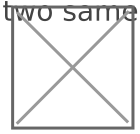



Again, this is mostly due to some online service resizing and lowering quality to ease on their bandwidth costs. There is usually no reason to keep the lower quality version.

same quality duplicates

When are two files the same quality? A good rule of thumb is if you scroll between them and see no obvious differences, and the comparison statements do not suggest anything significant, just set them as same quality.

Here are two same quality duplicates:

There is no obvious different between those two. The filesize is significantly different, so I suspect the smaller is a lossless png optimisation, but in the grand scheme of things, that doesn't matter so much. Many of the big content providers--Facebook, Google, Cloudflare--automatically 'optimise' the data that goes through their networks in order to save bandwidth. Although jpegs are often a slaughterhouse, with pngs it is usually harmless.

Given the filesize, you might decide that these are actually a better/worse pair--but if the larger image had tags and was the 'canonical' version on most boorus, the decision might not be so clear. You can choose better/worse and delete one randomly, but sometimes you may just want to keep both without a firm decision on which is best, so just set 'same quality' and move on. Your time is more valuable than a few dozen KB.

Sometimes, you will see pixel-for-pixel duplicate jpegs of very slightly different size, such as 787KB vs 779KB. The smaller of these is usually an exact duplicate that has had its internal metadata (e.g. EXIF tags) stripped by a program or website CDN. They are same quality unless you have a strong opinion on whether having internal metadata in a file is useful.

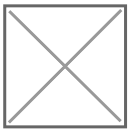
alternates

As I wrote above, hydrus's alternates system in not yet properly ready. It is important to have a basic 'alternates' relationship for now, but it is a holding area until we have a workflow to apply 'WIP'- or 'recolour'-type labels and present that information nicely in the media viewer.

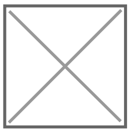
Alternates are not of exactly the same thing, but one is variant of the other or they are both descended from a common original. The precise definition is up to you, but it generally means something like:

- the files are recolours
- the files are alternate versions of the same image produced by the same or different artists (e.g. clean/messy or with/without hair ribbon)
- iterations on a close template
- different versions of a file's progress, such as the steps from the initial draft sketch to a final shaded version

Here are some recolours of the same image:



And some WIP:



And a costume change:

None of these are duplicates, but they are obviously related. The duplicate search will notice they are similar, so we should let the client know they are 'alternate'.

Here's a subtler case:

These two files are very similar, but try opening both in separate tabs and then flicking back and forth: the second's glove-string is further into the mouth and has improved chin shading, a more refined eye shape, and shaved pubic hair. It is simple to spot these differences in the client's duplicate filter when you scroll back and forth.

I believe the second is an improvement on the first by the same artist, so it is a WIP alternate. You might also consider it a 'better' improvement.

Here are three files you might or might not consider to be alternates:

These are all based on the same template--which is why the dupe filter found them--but they are not so closely related as those above, and the last one is joking about a different ideology entirely and might deserve to be in its own group. Ultimately, you might prefer just to give them some shared tag and consider them not alternates *per se*.

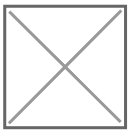
not related/false positive

Here are two files that match false positively:

Despite their similar shape, they are neither duplicates nor of even the same topic. The only commonality is the medium. I would not consider them close enough to be alternates--just adding something like 'screenshot' and 'imageboard' as tags to both is probably the closest connection they have.

Recording the 'false positive' relationship is important to make sure the comparison does not come up again in the duplicate filter.

The incidence of false positives increases as you broaden the search distance--the less precise your search, the less likely it is to be correct. At distance 14, these files all match, but uselessly:

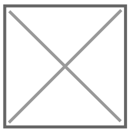


the duplicates system

(advanced nonsense, you can skip this section. tl;dr: duplicate file groups keep track of their best quality file, sometimes called the King)

Hydrus achieves duplicate transitivity by treating duplicate files as groups. Although you action pairs, if you set (A duplicate B), that creates a group (A,B). Subsequently setting (B duplicate C) extends the group to be (A,B,C), and so (A duplicate C) is transitively implied.

The first version of the duplicate system attempted to record better/worse/same information for all files in a virtual duplicate group, but this proved very complicated, workflow-heavy, and not particularly useful. The new system instead appoints a single *King* as the best file of a group. All other files in the group are beneath the King and have no other relationship data retained.



This King represents the group in the duplicate filter (and in potential pairs, which are actually recorded between duplicate media groups--even if most of them at the outset only have one member). If the other file in a pair is considered better, it becomes the new King, but if it is worse or equal, it merges into the other members. When two Kings are compared, whole groups can merge!

Alternates are stored in a similar way, except the members are duplicate groups rather than individual files and they have no significant internal relationship metadata yet. If α, β, and γ are duplicate groups that each have one or more files, then setting (α alt β) and (β alt γ) creates an alternate group (α,β,γ), with the caveat that α and γ will still be sent to the duplicate filter once just to check they are not duplicates by chance. The specific file members of these groups, A, B, C and so on, inherit the relationships of their parent groups when you right-click on their thumbnails.

False positive relationships are stored between pairs of alternate groups, so they apply transitively between all the files of either side's alternate group. If (α alt β) and (ψ alt ω) and you apply (α fp ψ), then (α fp ω), (β fp ψ), and (β fp ω) are all transitively implied.

Some fun. And simpler.

Revision #1
Created 12 March 2021 16:38:24 by CuddleBear
Updated 12 March 2021 18:24:07 by CuddleBear